GoDot

Graphics Manipulation & Conversion for the C-64 and C-128

END USER LICENSE AGREEMENT

REDISTRIBUTION NOT PERMITTED

GRANT.

Creative Micro Designs, Inc., ("CMD") hereby grants to you a non-exclusive license to use the Software and accompanying documentation ("Documentation") in the manner described under "Scope of Grant."

LIMITED WARRANTY.

CMD warrants that the media containing the Software, Documentation and other materials supplied with the Software, if provided by CMD, are free from defects in material and workmanship and will so remain for ninety (90) days from the date you acquired the Software. CMD does not warrant, however, that the Software or other materials supplied with the Software are fit for any particular use or that your use of the Software will be uninterrupted or that the operation of the Software will be error-free or secure. CMD's sole liability for any breach of this warranty shall be, in CMD's sole discretion: (i) to replace your defective media, Documentation or other materials normally supplied with the software; or (ii) to supply you with any portion of the Software, Documentation or other materials which CMD deems to be missing from the package as supplied to you by CMD. Repaired or replaced Software and Documentation shall be covered by this limited warranty for the period remaining under the warranty that covered the original Software, or if longer, for thirty (30) days after the date of shipment to you of the repaired or replaced Software. Only if you inform CMD of your problem with the Software during the applicable warranty period and provide evidence of the date you purchased a license to the Software will CMD be obligated to honor this warranty. CMD will use reasonable commercial efforts to repair or replace pursuant to the foregoing warranty within 30 days of being so notified.

THIS IS A LIMITED WARRANTY AND IT IS THE ONLY WARRANTY MADE BY CMD. CMD MAKES NO OTHER EXPRESS WARRANTY AND NO WARRANTY OF NONINFRINGEMENT OF THIRD PARTIES' RIGHTS. THE DURATION OF IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION, WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR A PARTICULAR PURPOSE, IS LIMITED TO THE ABOVE LIMITED WARRANTY PERIOD; SOME JURISDICTIONS DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO LIMITATIONS MAY NOT APPLY TO YOU. NO CMD DEALER, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS, OR ADDITIONS TO THIS WARRANTY. If any modifications are made to the Software by you during the warranty period; if the media is subjected to accident, abuse, or improper use; or if you violate the terms of this Agreement, then this warranty shall immediately be terminated. This warranty shall not apply if the Software is used on or in conjunction with hardware or software other than the unmodified version of hardware and software with which the software was designed to be used as described in the Documentation. **THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY BY JURISDICTION.**

SCOPE OF GRANT.

You may:

- use the Software on any single computer;
- use the Software on a second computer so long as only one copy is used at a time; or
- copy the Software for archival purposes, provided any copy contains all of the original Software's proprietary notices.

You may not:

- permit other individuals to use the Software except under the terms listed above;
- permit concurrent use of the Software;
- modify, translate, reverse engineer, decompile, disassemble (except to the extent applicable laws specifically prohibit such restriction), or create derivative works based on the Software;
- copy the Software other than as specified above;
- rent, lease, grant a security interest in, or otherwise transfer rights to the Software; or
- remove any proprietary notices or labels on the Software.

TITLE.

Title, ownership rights, and intellectual property rights in the Software shall remain in CMD and/or its suppliers. The Software is protected by the copyright laws and treaties. Title and related rights in the content accessed through the Software is the property of the applicable content owner and may be protected by applicable law. This License gives you no rights to such content.

TERMINATION.

The license will terminate automatically if you fail to comply with the limitations described herein. On termination, you must destroy all copies of the Software and Documentation.

EXPORT CONTROLS.

None of the Software or underlying information or technology may be downloaded or otherwise exported or reexported (i) into (or to a national or resident of) Cuba, Iraq, Libya, Yugoslavia, North Korea, Iran, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Denial Orders. By using the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list.

In addition, if the licensed Software is identified as a not-for-export product (for example, on the box, media or in the installation process), then the following applies: **EXCEPT FOR EXPORT TO CANADA FOR USE IN CANADA BY CANADIAN CITIZENS, THE SOFTWARE AND ANY UNDERLYING TECHNOLOGY MAY NOT BE EXPORTED OUTSIDE THE UNITED STATES OR TO ANY FOREIGN ENTITY OR "FOREIGN PERSON" AS DEFINED BY U.S. GOVERNMENT REGULATIONS, INCLUDING WITHOUT LIMITATION, ANYONE WHO IS NOT A CITIZEN, NATIONAL OR LAWFUL PERMANENT RESIDENT OF THE UNITED STATES. BY DOWNLOADING OR USING THE SOFTWARE, YOU ARE AGREEING TO THE FOREGOING AND YOU ARE WARRANTING THAT YOU ARE NOT A "FOREIGN PERSON" OR UNDER THE CONTROL OF A FOREIGN PERSON.**

LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL CMD OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL CMD BE LIABLE FOR ANY DAMAGES IN EXCESS OF THE AMOUNT CMD RECEIVED FROM YOU FOR A LICENSE TO THE SOFTWARE, EVEN IF CMD SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

HIGH RISK ACTIVITIES.

The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). CMD and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

MISCELLANEOUS.

If the copy of the Software you received was accompanied by a printed or other form of "hard-copy" End User License Agreement whose terms vary from this Agreement, then the hard-copy End User License Agreement governs your use of the Software. This Agreement represents the complete agreement concerning this license and may be amended only by a writing executed by both parties. **THE ACCEPTANCE OF ANY PURCHASE ORDER PLACED BY YOU IS EXPRESSLY MADE CONDITIONAL ON YOUR ASSENT TO THE TERMS SET FORTH HEREIN, AND NOT THOSE IN YOUR PURCHASE ORDER.** If any provision of this Agreement is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This Agreement shall be governed by Massachusetts law (except for conflict of law provisions). The application of the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.

U.S. GOVERNMENT RESTRICTED RIGHTS.

Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer-Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, or at 252.211-7015, or to CMD's standard commercial license, as applicable, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Creative Micro Designs, Inc., P.O. Box 646, 15 Benton Drive, East Longmeadow, MA, 01028-0646, USA.

Table of Contents

Quick Start Instructions 6
Section 1 - Starting Up 7
- System Requirements 7
- Getting Started 7
- CMD Device Users 8
- Pointer Navigation 8
- Selecting Function with Gadgets 8

Section 2 - The Main Screen 10
- Image Information 10
- Screen Controls 11
- Color Controls 11
- Load & Save Format 13
- Image Operators 13
- Commands Area 14

Section 3 - The File Requester 15
Section 4 - Supported File Formats 17
Section 5 - Processing Modules 25
- System Modules 25
- Generic Modules 28
- Error Distribution 29
- Convolution Modules 31

Section 6 - Printer Drivers 33
Section 7 - Tutorials 34
- Tutorial 1: Composing an Image 34
- Tutorial 2: Using Clips 35

Section 8 - GoDot Tips & Tricks 37
- Flipping an Image 37
- Creating Masks for ldr.4Bit&Mask 37
- Reconfiguring the REU 37
- svr.PFoxGB Produces Blocky Images 38
- Files Can't be Loaded 38
- FileCopy Tips 38
- PCX Pictures 38
- Files Can't be Saved 38
- DOS Function Refuses to NEW a Disk 38
- Umlauts via Keyboard 39
- Rendering Prepped Images 39
- INI Files 39
- INI File Command Summary 41

GoDot © 1991—1997 by A. Dettke & W. Kling

Distributed and Published in the United States by Creative Micro Designs, Inc.

Congratulations!

You've just purchased one of the most versatile and innovative programs ever released for the C-64 and C-128! If you experience media problems with the supplied software, contact CMD Technical Support by phone at 413-525-0023 (Mon.-Fri., 1PM-5:30 PM EST) or leave Email to *support@cmdweb.com*. For additional support for the program, visit the GoDot web site at *http://members.aol.com/howtogodot/welcome.htm*, or you may contact the authors directly via Email at *god.ADettke@t-online.de*. Also, be sure to check out the CMDWEB website at *http://www.cmdweb.com/* for all the latest in Commodore computing!

— TO LOAD GoDot —

```
LOAD"GODOT",8 (RETURN)
```

with JiffyDOS-equipped systems you may also use:

```
   ↑GODOT (RETURN)
or ↑"GODOT",8 (RETURN)
```

Note: You may susbstitute a different device number for the "8" in the commands above, but be aware that GoDot will only load properly from devices 8, 9, 10 or 11. For additional details on starting and using GoDot, see "Starting Up" on the following pages.

SECTION 1
STARTING UP

System Requirements

GoDot was written so that it may be used on any Commodore 64/128 with a minimum configuration. However, the program is capable of taking advantage of the latest technology available for the Commodore, and will run more efficiently on systems with more powerful peripherals such as RAM expanders, high capacity storage devices and/or acceleration.

Minimum Requirements: Commodore 64, TV/Monitor, 1541 Disk Drive

The following items are also supported by GoDot, and some will provide additional enhancements not available with a minimal system:

Computers:	Commodore 64c, Commodore SX-64, Commodore 128, Commodore 128D
Input Devices:	Commodore and compatible Joysticks, Commodore 1350 JoyMouse, Commodore 1351 Mouse, CMD SmartMouse
Storage Devices:	Commodore 1541-II, Commodore 1541C, Commodore 1571, most 1541 or 1571 compatible drives, Commodore 1581, CMD FD-2000, CMD FD-4000, CMD HD Series Hard Drives, CMD RAMLink
RAM Expansion:	Commodore 1700 REU, Commodore 1764 REU, Commodore 1750 REU, SSI SuperClone 1750 REU, CMD 1750 REU, CMD 1750XL REU
Printers:	Epson and compatible 9-pin and 24-pin black & white or color printers, Cannon BJ-series Bubble Jet printers, HP DeskJet and DeskJet-compatible printers, most PCL3-compatible printers (including laser printers), Epson Stylus series ink-jet printers
Other:	CMD SuperCPU Accelerator, Scanntronik Handyscan 64, Scanntronik Video Digitizer

Getting Started

Before you begin, we recommend that you make a backup copy of your original GoDot disk. The distribution disk is not copy protected, so you may copy it using any standard Commodore file or disk copier. Please do not abuse the fact that the program may be easily copied; this was done to make the program more flexible for your use, and not to assist you in distributing illegal copies to others. GoDot is a copyrighted product and is protected under international copyright laws. You are entitled to make a backup for your own archival use, or install a copy of the files for use and put the original away as your backup. As indicated in the license agreement, you are entitled to have only one copy of the program in use at any given time.

To load GoDot, insert your backup copy of GoDot into your disk drive which is configured as device 8 and type:

```
LOAD"GODOT",8
```

After typing this, be sure to press the (RETURN) key. After the program has successfully loaded and you see the READY prompt, type:

```
RUN
```

Again, be sure to press the (RETURN) key after typing this command to start the program. *Note: You may use other device numbers by replacing the number 8 in the LOAD command with a different device number; GoDot supports booting from devices 8, 9, 10 or 11.*

CMD Device Users

Several modules are included with GoDot to provide support to CMD device users. However, due to the way GoDot accesses data on disks it will always exit to the root directory of a partition on startup. If you install GoDot in a subdirectory, be sure to place a copy of *mod.ChangeDir* in the root directory. After booting, ***load*** and ***execute*** *mod.ChangeDir*, using it to regain access to the subdirectory containing your other GoDot files.

Pointer Navigation

GoDot accepts three forms of input: mouse, joystick or keyboard. Although a mouse or joystick is preferable, you can move the pointer with the cursor keys. Movement using the cursor keys can be somewhat slow and for this reason, GoDot uses (COMMODORE) + (S), (COMMODORE) + (F), (COMMODORE) + (H), and (COMMODORE) + (F3) to achieve faster movement.

When moving the pointer on the screen, you will notice that the pointer can be moved into the border area. This gives you the ability to hide the pointer when viewing your work using the Redisplay (***Redisp***) option.

Selecting Functions With Gadgets

The main screen contains a number of buttons, or *gadgets*, which can be used to select different functions. To select the function provided by a particular gadget, move the pointer using the mouse, joystick or cursor keys. When the upper left portion of the pointer is resting over the gadget you wish to select, press the (RETURN) key, the joystick fire button, or the left mouse button.

To the right of your screen, you'll find a gadget labeled ***GoDot!***. Selecting this gadget provides you with information about the program, such as the version number and the names of the programmers.

Selecting the ***Redisp*** gadget lets you view the current image without making any alterations to it. Please note that it is normal for the screen you see in Redisplay mode to contain random garbage if you have not yet loaded a picture. To exit from Redisplay mode, press the (RETURN) key, the joystick fire button, or the left mouse button.

The other gadgets located on the main screen will each open sub-screens, or requesters, which reveal various other functions and features. For example, to see a listing of the contents of the current floppy disk, select the ***Load*** gadget. The requester that opens is referred to as a ***File Requester*** and will be covered in detail later in this manual. For now, select the ***Cancel*** gadget to return to the main screen.

To exit GoDot you would select the ***Exit*** gadget. At this time we will ask you to select that option so as to demonstrate yet another type of requester. Selecting ***Exit*** will open a small requester in front of the main screen. When a requester like this is opened, you may only access the gadgets found within that requester. Occasionally you may find a situation where one requester opens on top of another. In that situation, only gadgets on the frontmost requester is active. All requesters are equipped with at least one gadget labeled ***Cancel*** or ***Leave*** which will allow you to go back to the previous requester or screen.

Some Gadgets have the ability to switch between two or more different options. These are called *Cycle Gadgets*, and they show which option is currently selected for a function. For example, the ***Resolution*** gadget in the *Screen Controls* area lets you switch between the two available screen modes: *Hires* and *Multi*. The gadget next to the Dith: label (the ***Dith*** gadget) offers an even more extensive list of options.

In the lower left corner of the main screen you'll find a gadget labelled ***Display***. Try it! When you do, you'll see a display of random graphics since no image has been loaded. If there were an image in memory, GoDot would take time to scan through all the gadgets to collect any changes which need to be made to the image. These changes would then be made before displaying the picture. The Display function can be very slow on unaccelerated systems. If you wish to interrupt the Display function, you may do so at any time by pressing the (STOP) key.

In the upper right corner of the screen is a gadget called ***Redisp***. This function lets you view the image as it was shown when you last used the ***Display*** gadget. It will not recalculate any changes you have made. This is useful when you wish to view the image while deciding on what changes you would like to make. ***Redisp*** is a much faster way to view an image because it simply switches between the main screen and the image that is presently in memory.

Let's try to view an image. Click the ***Load*** gadget in the upper right corner of the main screen. Once the requester comes up, press the ***Replace*** gadget. On the left hand side of the screen you'll see a directory which lists a variety of images. You may scroll up and down this list by using the arrows positioned in the middle of this screen. For this exercise we suggest that you select the file named *Pinball.4bit*, however, any file ending in *.4bit* will do. To select the file to be loaded, click once on the name and select ***Load***, or simply double-click on the name of the desired file. The status bar (which is displayed across the bottom of the screen) shows the progress of the load. After loading the file, GoDot will switch back to the main screen. Now, you may hit the ***Display*** gadget to see the file that has just been loaded.

This completes our tour of the main screen. Refer to the sections that follow for additional information on individual functions and features of the GoDot image processing system.

SECTION 2

THE MAIN SCREEN

The main screen is divided into six separate areas which helps to group similar functions. Each area has one or more *gadgets*, which are equivalent to push-buttons. Functions are selected by placing the pointer over the associated gadget and pressing the (RETURN) key, or by pressing or the left button of the mouse or the joystick fire button. In some cases this may lead to an immediate action; in other cases the label may change, or parameters may be modified. If a gadget has more than two options, you may move forward through the options by clicking on the right side of the gadget button, or backwards by clicking on the left side of the gadget.

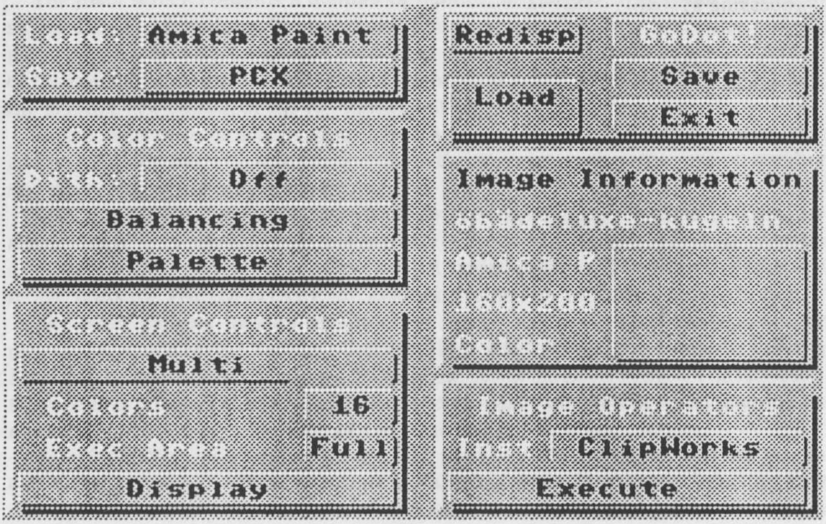
A good example of this feature can be seen in the ***Dith*** gadget. If you click on it you will find that there are numerous options and by clicking to the right and left sides within the button, you can scroll back and forth through them.

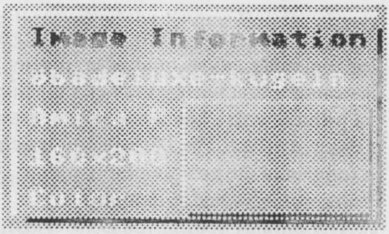
The upper two areas of the main screen have no specific labels. Throughout the balance of this manual they will be referred to as the *Install* area (on the left) and the *Commands* area (on the right).

Image Information

The Image Information area contains the largest gadget found on the main screen, the ***Preview*** gadget. When activated by a click it shows a tiny preview of the image buffer in four shades of gray. It is deactivated by another click.

The ***Preview*** gadget is also used by many of the GoDot modules which do not open their own requesters to show their rate of progress. In this case a thin white





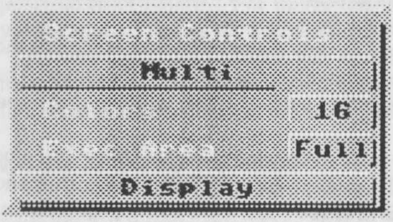
progress line moves down through the gadget as the module makes progress.

The Image Information area also contains information about the currently loaded image. When you begin working with GoDot, this area will simply report that the current document is "Untitled" until an actual image has been loaded. If you hit this gadget before loading an image, you may find random character garbage.

Once an image has been loaded, all necessary information will appear above and to the left of the preview gadget This information includes the file name, format, resolution and color mode.

Screen Controls

The gadgets found within this area control the attributes of the image that is currently in the image buffer. In other words, these gadgets control the parameters which dictate how the image will be displayed.



Within the *Screen Controls* area, there are four gadgets: ***Resolution***, ***Colors***, ***Exec Area*** and ***Display***. Each helps to control how images are displayed on the screen.

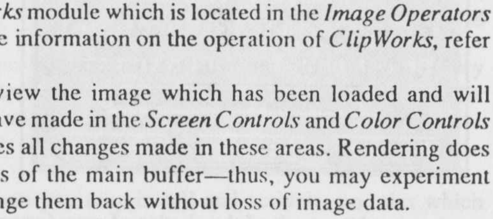
The ***Resolution*** gadget switches between *Hires* (320 x 200 pixel resolution) and *Multi* (160 x 200 pixel resolution). Obviously, *Hires* will give you a higher degree of clarity. However, *Multi* gives you more colors at a lower resolution. When using monochrome monitors, we recommend using *Hires* mode. It is important that the proper screen resolution is selected before exporting to specific C64 image formats, as certain programs will have specific resolution requirements.

The ***Colors*** gadget allows you to specify the number of colors used while rendering, from monochrome (2 colors) to the full 16 color range. Note that you may adjust the colors via the ***Palette*** in the *Color Controls* area of the main screen.

The ***Exec Area*** toggles between *Full* and *Clip* modes. These modes display either the entire image (*Full*) or a selected portion of the image (*Clip*). The size of the clip may be adjusted using the *ClipWorks* module which is located in the *Image Operators* area on the Main Screen. For more information on the operation of *ClipWorks*, refer to Section 5.

The ***Display*** gadget lets you view the image which has been loaded and will implement any changes that you have made in the *Screen Controls* and *Color Controls* areas. While rendering, GoDot uses all changes made in these areas. Rendering does not, however, change the contents of the main buffer—thus, you may experiment with the control gadgets, and change them back without loss of image data.

Color Controls



The *Color Controls* section of the main screen houses the features which influence the color and intensity settings used for a certain image. There are three gadgets found in this area: ***Dith*** (Dither), ***Balancing***, and ***Palette***.

The ***Dith*** gadget (Dith is an abbreviation for the term dithering) provides a means for replacing colors or shades of gray with a dot pattern. Dithering is recommended for images that will be viewed on a monochrome monitor, or for images which are to be printed with a black and white printer. Dithering can also be used to simply create a desired effect. The ***Dith*** gadget is actually a button. Each time the button is pressed, it toggles to the next available option. There are five dithering settings: *Off*, *Ordered*, *Pattern*, *Noise* and *Random*.

The default setting for ***Dith*** is *Off*, since you may often be working with color images, and relatively few users will use GoDot with a monochrome monitor. In the *Off* position, darker colors will be black and lighter ones will be white (assuming you have chosen the default black & white palette for the two colors used).

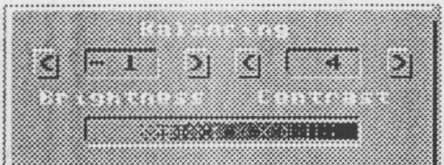
The second option is *Ordered*, whose primary purpose is to display pictures on monochrome displays. It avoids series of dots which cannot be displayed in the same way on a monochrome screen as they exist in memory (i.e., vertical one dot lines.)

Pattern is a dither type that was created especially for output to printers. On screen, it looks rough and incomplete; however, on paper it is finely structured, and shades of gray are very well approximated. The ***Pattern*** dither type is the only dithering style that can be edited (using the module *.PatternEd*). If you want to send the picture to a black & white printer, ***Pattern*** would be the best choice.

Both *Ordered* and *Pattern* are very similar. The only major difference is the fact the you can edit *Pattern*, allowing you to create special effects or optimize the output. The *.PatternEd* module is also used to create masks for *ldr.4Bit&Mask*, which is discussed later in the manual.

The *Noise* dithering type is quite unique, and can be used to create some very nice effects. *Noise* replaces gray scale or color with a random dot pattern that will correspond to the actual brightness. Thus, an image that is dithered with *Noise* will appear different each time *Noise* is applied. *Noise* can best be adjusted by using additional contrast, and the results of this may look something like a charcoal drawing.

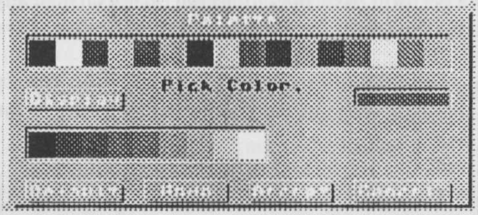
Random works much like *Off* when ***Colors*** is set for a level of 2. However, with *Random* the rendering process is noticeably faster. ***Random*** was originally developed as a way to mark areas of interest (using Brightness and Contrast to control which areas). This is often useful for creating a '***mask***', or an area or areas which are (or are not) to be affected by further editing.



The ***Balancing*** gadget opens a new requester which provides you with functions similar to those found on your TV or monitor. You may control the ***Brightness*** and ***Contrast*** of the displayed image by selecting the appropriate arrow to the right or left of the value displayed for these parameters. The visual display on the requester helps you to see how each of the shades of gray (or color) within your image are affected by the adjustment—each shade is represented by a dot pattern, with the solid end of the display being the darker color.

There are three gadgets found on the *Balancing* requester: ***Reset***, ***Accept*** and ***Cancel***. The ***Reset*** gadget resets both the ***Brightness*** and ***Contrast*** to zero. The ***Accept*** gadget will implement any changes you've made, while ***Cancel*** lets you exit the requester without making any changes. *Note: Some modules will reset the Balancing values automatically, while others may use them or set them for their own purpose.*

The ***Palette*** gadget opens a new requester containing several gadgets which are used to compose a color palette for rendering. The number of colors is specified by the ***Colors*** gadget in the *Screen Controls* area of the main screen. The *Palette* requester presents two lines of colored squares. The top line, called the *Source Palette*, shows the 16 colors from which you can choose; the bottom line, called the *Destination Palette*, is the new palette that you are creating. If you select a color from the *Source Palette*, it will be shown in a special box we call *Last Picked*, found at the right center of the *Palette* requester. By clicking on a square in the *Destination Palette*, you will assign the *Source Palette* color shown in the *Last Picked* box to that location in the *Destination Palette*, thus making the change complete.



If you opened this requester by mistake or decided that your palette should remain the same as it was prior to opening it, you may select ***Cancel*** to exit from the *Palette* requester. If you make a color change, but then change your mind, use the ***Undo*** gadget undo your last modification without exiting.

The ***Default*** gadget automatically suggests a color combination for each of the 14 possible color palettes. For image data loaded as 16 gray scales, these palettes give you a range of colors for use in achieving colorizing effects.

To view the color changes that you have made without having to go back to the main screen, we've implemented a ***Display*** gadget in the *Palette* requester. If you like what you see, changes are confirmed by selecting the ***Accept*** gadget.

Load & Save Format

The small area in the upper left corner of the main screen contains two long gadgets with the text **Load:** and **Save:** next to them. These gadgets contain the current file formats for disk input and output. Clicking on either the ***Load*** or ***Save*** gadget, will open the GoDot file requester for selecting a different loader or saver module. These modules are available for nearly all C64 image formats, and for many formats used by other computers such as Atari ST, Commodore Amiga or MS-DOS/Windows. On disk, these loader and saver files have the prefix *ldr.* or *svr.* Through these modules, GoDot becomes a powerful conversion utility for converting images to and from your favorite Commodore graphics program, as well as for exporting to (or importing from) other computer platforms. Note that it is very important to select the correct loader or saver prior to loading or saving a particular file type from the *Commands* area on the main screen.

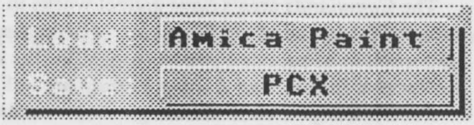
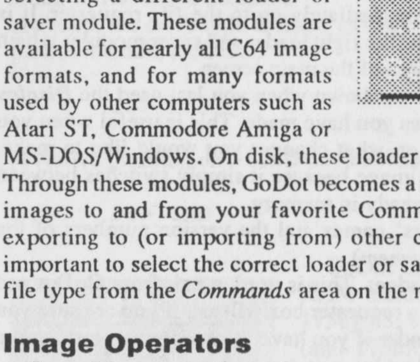


Image Operators

From this area on the main screen you may install and activate modules which enhance the capabilities of GoDot or allow you to perform special tasks. These modules range from image processing modules to a host of tools for controlling peripherals and internal parameters.

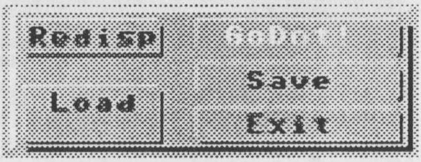


Clicking on the ***Inst*** gadget (which shows the name of the currently installed module) will open a file requester containing files with the *mod.* prefix. After selecting one of these files, clicking on the requester's ***Load*** gadget will load the selected module. For most modules you must then select ***Execute*** to launch the module, though there are some exceptions (such as *mod.DecodeGIF*, which simply needs to be loaded in order to use the *ldr.GIF* loader). When you execute a module, it may open a new requester to provide options. Module activity can often be seen with the white ***progress line*** in the ***preview*** gadget.

Commands Area

The gadgets found within this area perform basic operations but do not change any parameters. There are five gadgets: ***Redisp*** (short for redisplay), ***Load***, ***Godot!***, ***Save*** and ***Exit***.

Load lets you select a picture file from the file requester. In some cases, certain loader types will open a requester that is specific to that loader before going to the file requester. When you press the ***Load*** gadget, you will activate the *Load* requester, in which you'll find four gadgets: ***Source***, ***Replace***, ***Compose*** and ***Cancel Load***. Next to the **Source:** label is the ***Source*** gadget which displays the selected RAM device (if any). Clicking on this gadget moves you through the available RAM devices.



Use the ***Replace*** gadget when you wish to load a picture into memory and have it replace any other picture that is already in the computers memory. The ***Compose*** gadget is used to apply certain types of graphics, i.e. foregrounds, backgrounds etc. For more info on this gadget see Section 7.

The last gadget is ***Cancel Load***. As its name indicates, this gadget will let you cancel prior to starting the load process, and exit from the ***Load*** requester without affecting the image currently stored in memory.

The *Commands* area's ***Save*** gadget works in a manner similar to the ***Load*** gadget, except that by selecting Save, you will immediately go to the file requester. It is important to be sure that you have installed the right loader and saver modules which are located in the *Load & Save Format* area of the main screen.

Redisp lets you view the image as it was shown when you last used the ***Display*** gadget. It will not recalculate any changes you have made. This is useful when you wish to view the image while deciding on what changes you would like to make. ***Redisp*** is a much faster way to view an image because it simply switches between the main screen and the picture that is already in memory.

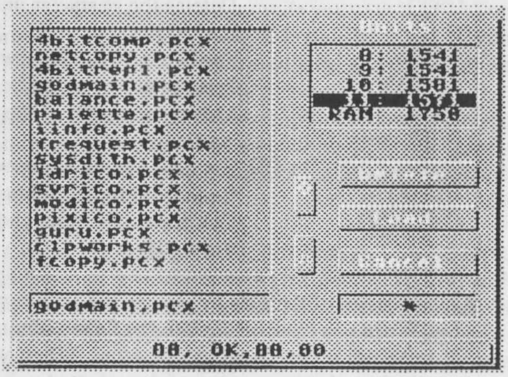
GoDot! provides you with the authors' names and the version numbers of the main program files (*god.main* and *god.upmem*).

Last but not least, we have the ***Exit*** gadget. This is used to exit from GoDot and return to BASIC. When you select ***Exit***, a requester box will ask if you are sure you want to quit. This can be a helpful reminder if you haven't saved your work, and a safety factor in case you hit the ***Exit*** gadget by mistake.

SECTION 3

THE FILE REQUESTER

The *File Requester* is the core of GoDot's user interface. From it you control all storage devices which are connected to your system: floppy disk drives, RAM expansion units, RAMLink and even hard disk drives.



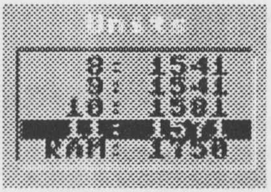
```
4bitcomp.pcx
netcopy.pcx
4bitrep1.pcx
godmain.pcx
balance.pcx
palette.pcx
info.pcx
frequest.pcx
sysdith.pcx
ldrico.pcx
svrico.pcx
modico.pcx
pixico.pcx
guru.pcx
clpworks.pcx
fcopy.pcx

godmain.pcx
*
08, OK,00,00
```

In the upper-right corner of the *File Requester* you'll find the *Units* window. During start-up, GoDot checks all peripherals connected to the system and lists them in this window. GoDot supports all common drive types (1541, 1571, and 1581). If a drive cannot be identified, GoDot assumes it is a 1541. CMD devices are listed with a greater-than symbol (>) following the device number instead of a colon (:). Also, when CMD Native partitions are used, the drive type is listed as *CMD*. To illustrate, a CMD device with a 1581 partition would look like:

```
10> 1581
```

A Native partition on a CMD device would be listed as:

```
10> CMD
```



GoDot also recognizes many types of additional RAM including: Commodore REUs (1700, 1764 and 1750), SSI's Super 1750 Clone, GEORAM, CMD 1750 (and 1750XL) REUs, CMD's RAMLink, 16K VDC RAM (C-128), 64K VDC RAM (C128 with upgrade or C-128D) and the PageFox DTP cartridge (which can be accessed via the *Temp* saver). If no additional memory (RAM) is found, the type is displayed as *C64* (or *C64n* for the newer models).

The current drive can be changed by clicking a unit number. GoDot will read the directory and display the file names in the *File Listing* window on the left side of the *File Requester*. If you have more than one drive installed, GoDot will keep track of the units that you have used during the current session for loading, saving, and installing modules. *(Note: Only those files matching the settings of the initiating function will be displayed. The filter that is currently in use is shown in the window* *below the **Cancel** gadget. For example, if the file requester is invoked by the* PatternEd *module, only pattern files with the prefix* pat. *will be shown in the file box at the left.)*

You may scroll up and down through the files listed in the *File Listing* by using the ***Up*** and ***Down*** arrow gadgets located in the center of the *File Requester*.

Located below the *File Listing* window (in the lower-left area of the *File Requester*) is the *Input* window. By clicking once on a file name in the *File Listing* window, the name will be transferred into the *Input* window. To load the selected file you can click on the ***Load*** gadget found on the right side of the *File Requester*. A file may also be loaded by simply double-clicking on the file name in the *File Listing* window.

You also have the ability to edit the file name displayed in the *Input* window. To edit a file name, place your pointer over the input window and click on it; a cursor will appear in the window where you clicked. From the *Input* window you have access to the following editing functions:

DEL	Deletes the character under the cursor (different than BASIC)
INST	Toggles between Overwrite and Insert modes
HOME	Sets the cursor to the beginning of the Input window
CLR	Moves the cursor to the end of the string in the Input window
COMMODORE + X	Clears the Input window
STOP	Exits from Input without changes
RETURN	Completes the input and sets the pointer to the Load or Save gadget

At the bottom of the *File Requester* is a long rectangular gadget called the *status display*. When you click on the ***Load*** or ***Save*** gadgets, this area will display a progress bar to track the progress of the operation. When you aren't loading or saving, this gadget displays the error status of the selected unit.

The ***Delete*** gadget may be used to remove a selected file from the disk. *Note: This action is not reversible! If you do delete a file by mistake, you'll need to exit from GoDot without writing anything else to the disk and try to recover the file using an unscratch utility or (if you're very knowledgeable and capable in such matters) a track and sector editor.*

SECTION 4
SUPPORTED FILE FORMATS

GoDot supports a wide variety of different image formats. The focus of this section is to provide the general specifications needed to use these different formats. For each of the major format types, you will find a list of the loaders and savers that are designed to be used with that format type. Additionally, you will find notes and suggestions to help you use the format. Even though we provide you with information regarding a given format, it is probably best if you simply take the time to play with it. You'll find that most of the loaders and savers work similarly, and once you've learned to use one of them you can generally operate any of them.

Due to the wide variety of image data formats, GoDot attempts to convert all image data to its own *4Bit* format when loading. The *4Bit* format features 320 x 200 pixels with the standard Commodore 16 colors and offers superior resolution and color selection. If you are loading a color image as a gray scale image, GoDot will interpret the colors as different shades of gray.

While GoDot can load most Commodore file formats, but foreign formats often require more colors and/or resolution than the Commodore can display. To deal with this, GoDot includes a set of sophisticated yet easy-to-use loader modules that let you select a part of the image or re-size it. As for the colors that cannot be displayed, GoDot will automatically assign the closest stock Commodore colors possible. In most cases, the outcome is very impressive.

To control the loader options, some loaders will open an options requester before invoking the file requester. Likewise, some savers will display options after choosing the output file name.

You may recall the ***Load*** gadget from earlier sections of the manual. When this gadget is clicked, a sub-menu appears listing all of the load modules found on the default drive. These loaders are actually input filters which allow GoDot to read different types of images. Since the Commodore never really had a graphic standard, many programs will use a unique format to store data. Sometimes these formats save the graphical data in a packed form.

The remainder of this section will consist of the technical specifications of the individual loaders and savers that are compatible with a given format along with additional notes and tips for its use.

4 Bit-Format

Format: 4BitGoDot (GoDot's native Format)
Displayed Info: 4BitGoDot, 320x200, Gray
Loaders: ldr.4BitGoDot *ldr.4BitGoDot* is the default loader module. There are special versions for composition and other manipulations during the loading process. These are referred to in the tutorial section.
ldr.4Bit&Arith
ldr.4Bit&Map
ldr.4Bit&Mask
ldr.4BitRGB(d)

ldr.RawData — Ldr.RawData expects 32000 bytes of 4Bit image data, making it easy to create files on other platforms without using the GoDot file header format and compression technique

Savers: svr.4BitGoDot
svr.Morph!64
svr.Temp

Notes: Files appear in the format, *name.4bit.* No special Temp loader is needed, because all 4Bit loaders support the temp file as input.

AFLI

Format: Hires FLI, uncompressed (65 blocks)

Displayed Info: Hi-Fli, 320x200, Color

Loaders: ldr.AFLI

Savers: svr.AFLI

Notes: Saving in AFLI format is good for 320x200 hires data images loaded with the IFF loader module. IFF and PCX are good formats to convert AFLI images into.

Amica Paint

Format: Multicolor, compressed

Displayed Info: Amica P, 160x200, Color

Loaders: ldr.Amica Paint

Savers: svr.Amica Paint

Notes: This is a compressed Koala file with an additional 256-byte color cycle list.

Basictext

Format: BASIC program containing PRINT statements or REM commands

Displayed Info: Text, 320x200, Color

Loaders: ldr.BasicTxt

Savers: none

Notes: This loader converts PETASCII text contained in BASIC programs into 4Bit data, using the following criteria: strings within quotes in PRINT statements are interpreted and placed as data, while commands for aligning data or changing colors are specified in quoted strings within REM commands. REM commands are interpreted as follows: The loader module interprets 3 numbers following an initial quote. The first character specifies alignment (0=align left [default], 1=align right, 2=align centered); the second is a hexadecimal number (0 through f) which selects a character color (default is 1); and the third character is a hexadecimal number which selects the background color of the character. With the latter, any non-hexadecimal character specifies that the background should be transparent, which is the default. With the REM command, a colon (:) can be used to signify the end of input, allowing you to retain comments within the REM command. You may mix PRINT and REM as you like, but note that the GoDot character set contains umlauts (special German characters). The following is an example of using REM statements within a BASIC program to define attributes:

```
REM "220:This is the header      (centered, red on black)
REM ":This is the first line     (same attributes as prev.)
```

Blazing Paddles

Format: Multicolor, uncompressed (40 blocks)
Displayed Info: BlPad, 160x200, Color
Loaders: ldr.BlPaddles
Savers: none
Notes: C64 color file format, the filenames contain a *PI.* prefix.

Botticelli (Plus4)

Format: Multicolor, uncompressed (40 blks)
Displayed Info: n/a
Loaders: none
Savers: Plus4MC16 Plus4 file format using the 121 native Plus4 colors. Although you can save your work using the C64's colors (***Color C64*** option), some amazing results can be achieved using your own Plus4 color palette previously made with the help of the *p4pmaker.ted* utility (***Custom Palette*** option). You may also use the 9 gray scales of the Plus4 color set (***Gray 9***).
Plus4MC121 This saver assumes you have imported data by using a special loader (*ldr.PCXprep4Pl4*) and will save a picture using up to 121 colors.
Notes: In this very special case we actually have 7-Bit data. Performing manipulations with standard modules will generally cause unexpected (and often undesired) results.

Degas (Atari)

Format: Hires, 640x400, uncompressed (127 blks)
Displayed Info: Degas, 320x200/160x200, Color
Loaders: ldr.Degas.Pl3
Savers: none
Notes: Atari ST monochrome 640x400 file format. During the loading procedure the data will be converted using a resampling method resulting either in five gray scale 320x200 segments, or nine gray scale 160x200 segments.

Diashow

Format: Multicolor, hires or text, compressed
Displayed Info: Diashow, 320x200/160x200/Text, Color
Loaders: ldr.Diashow
Savers: none
Notes: File format of a very flexible hardware framegrabber module. GoDot recognizes all imported data from the file except sprite data.

Doodle

Format: Hires, uncompressed (37 blks)
Displayed Info: Doodle, 320x200, Color
Loaders: ldr.Doodle
Savers: svr.Doodle
Notes: The classic color hires drawing program. Filenames begin with *DD*.

FLIP

Format: FLI, compressed/uncompressed (68 blks)
Displayed Info: FLIP, 160x200, Color

Loaders: ldr.FlipRaw
Savers: svr.FlipRaw
Notes: FLIP is an FLI paint program with all standard and many special features. The crunched file format is not supported.

Funpaint II

Format: IFLI, compressed/uncompressed (133 blks)
Displayed Info: IFLI, 160x200, Color
Loaders: ldr.IFLIprep4PCX
Savers: svr.FunPaintII
Notes: The file format of Fun Painter's successor, GunPaint, is also recognized. Interlaced FLI is theoretically capable of displaying 128 colors. The main drawback is the extensive flicker effect caused by many of the color combinations. GoDot uses only 55 colors, all of them chosen by significant difference and minimized flicker. Nevertheless, some necessary colors do still cause flicker. You can reduce this by preparing the output with *mod.FlickerFixer*. IFLI images are a good choice for exporting to other platforms, especially the PCX format using 256 colors (*svr.PCX-VGA*). To use IFLI for C64 multicolor output, prep the data with *mod.RasterIFLI*.

GeoPaint

Format: Hires 640x720, compressed
Displayed Info: n/a
Loaders: none
Savers: svr.GeoPaint
Notes: The data is saved as a 640x400 monochrome VLIR file. Choose your favorite dither type in the *Color Controls* section of the main screen. Raising the contrast settings is also recommended in most cases.

GIF (MS-DOS)

Format: IFLI, up to 1024x768, compressed
Displayed Info: GIF xxx, 160x200, Color
Loaders: ldr.GIF
Savers: none
Notes: To launch this loader module *mod.DecodGIF* must be installed first. Due to the quite sophisticated and memory hungry decoding routines, the loader section provides the user interface and then invokes the decoding procedures. The loading requester customizes GIF 87a and 89a pictures up to 1024x768 pixels in size with support of the full 256 color specifications. There is one restriction: interlaced pictures are not supported (yet). The largest area shows the preview shape of the source picture titled by the GIF format signature; the size and color depth are displayed in separate gadgets. The SetClip gadget lets you place a frame to set the section to be imported. The left and top offset is then displayed (also in specific gadgets). Rather than enlarging the clip selector frame, the ***X-Skip*** and ***Y-Skip*** gadgets make the preview shape appear smaller (or larger when setting double size). The gadget right under the ***SetClip*** toggles the color assignment modes from *Color* (more than 50 simulated colors) to *Gray 9* (for gray screen output) to *Gray 16* (for printing). The ***Level*** gadget controls the assignment of gray scales.

Handyscanner

Format: 4Bit, 640x512 (640x400 effectively)
Displayed Info: GoDotScan/HScan, 320x200, Gray
Loaders: ldr.Handyscanner
Savers:
Notes: The Scanntronik scanner hardware is supported directly. Select ***Load*** in the commands area and scanning is activated—even if the hardware is not present. You may abort by hitting the STOP key, loosing the actual contents of the 4Bit memory. The maximum scan area is approx. 5x3" (8x5 cm) using the highest scan quality (1:1), which corresponds to the 300% setting of the original software. The dithered scan input is converted via resampling and results in a 5-level gray scale hires picture.

Hires Bitmap

Format: Hires, uncompressed (32 or 33 blks)
Displayed Info: HiBitMap, 320x200, Color
Loaders: ldr.HiBitMap
ldr.PFoxSelect
Savers: svr.Bitmap
Notes: Composing multiple bitmaps is supported in a manner similar to the function of the *ldr.4BitGoDot* loader. As no color information is provided, you may set the color in the palette requester. Images loaded by *ldr.PFoxSelect* must be inverted using *mod.Negative*, as Scanntronik image material always comes up black on white.

Hires Manager

Format: FLI hires, compressed/uncompressed (65 blks)
Displayed Info: HiresM, 320x200, Color
Loaders: ldr.HiManRaw
Savers: svr.HiManRaw
Notes: The file names of this hires FLI editor begin with "(()". Only uncompressed images are accepted.

IFF ILBM (Amiga)

Format: Hires, compressed
Displayed Info: IFF xxx, 320x200, Color/Gray
Loaders: ldr.IFF
ldr.IFFdither
Savers: svr.IFF
Notes: GoDot supports all sizes up to 1024x768, and all resolutions up to 6 bit-planes including EHB and Ham6 modes. The picture will be scaled down automatically for the correct aspect. Since colors must be reduced, you may cut down the allocation of grays by adjusting the gray level from 0 (all) to 5 (colors only) using the ***Lvl*** gadget. To prepare for printer or gray scale output choose "Gray Scale".

Koala

Format: Multicolor, uncompressed (40 blks)
Displayed Info: Koala, 160x200, Color
Loaders: ldr.Koala
Savers: svr.Koala
Notes: Possibly the most common C64 graphic file format, Koala images recognized by most native graphics applications as long as the

filename is exactly 15 chars long, with trailing spades after the name text until the final three characters which are *PIC*.

Magic Disk EMC

Format: FLI, uncompressed (68 blks)
Displayed Info: MD-EMC, 160x200, Color
Loaders: ldr.MagicDiskEMC
Savers: none
Notes: Another 160x200 FLI editor distributed by the German disk mag *Magic Disk*.

Mandelbrot Construction Set

Format: FLI, compressed
Displayed Info: MCS-EMC, 160x200, Color
Loaders: ldr.M.C.S.
ldr.M.C.S.dat
Savers: svr.MCSfli
Notes: A good program for producing fractal images, even in 3-D. Capable of producing FLI output and storing an internal 256 color format.

Multicolor Bitmap

Format: Multicolor, uncompressed (32 or 33 blks)
Displayed Info: MCBitmap, 160x200, Color
Loaders: ldr.MCBitmap
Savers: ldr.Bitmap
Notes: 4 color format, but without color information.

OCP Art Studio & Advanced OCP Art Studio

Format: Hires, uncompressed (36 blks)
Multicolor, uncompressed (40 blks)
Displayed Info: OCP Art, 320x200/160x200, Color
Loaders: ldr.OCPArtStudio
Savers: none
Notes: Powerful hires (multicolor) paint program from England. All file names end with *PIC*.

Pagesetter (Plus4)

Format: Hires 440x720, compressed
Displayed Info: PSetter, 320x200, Color
Loaders: ldr.PSetSelect
Savers: svr.Pagesetter
Notes: This is a port of the C64 DTP program *Pagefox*. For information about the loader, refer to *ldr.PFoxSelect*. The saver performs a clockwise rotation providing a 400x600 pixel monochrome dithered image centered on a typical page.

Paint Magic

Format: Multicolor, uncompressed (37 blks)
Displayed Info: PMagic, 160x200, Color
Loaders: ldr.PaintMagic
Savers: none
Notes: Although it uses multicolor mode, this aged paint program used color RAM like an ordinary color register.

PCX (MS-DOS)

Format: Hires, compressed
Displayed Info: PCX-EGA, 320x200, Color/Gray
PCX-VGA, 160x200, Color/Gray
Loaders: ldr.PCX-EGA
ldr.PCX-VGA320
ldr.PCX-VGA320d
ldr.PCXprep4IFLI
ldr.PCXprep4PI4
Savers: svr.PCXmono
svr.PCXmono.90
svr.PCX-VGA
Notes: As nearly all paint programs on MS-DOS PCs are able to save images in this format, GoDot provides different loaders and savers to cover as many variations as possible, from monochrome clip art to 256 color pictures. While "*mono*" writes b/w 640x400 pixels, "*EGA*" uses all 16 colors in 320x200 pixel format. When converting from FLI format use the "*VGA*" saver. Pictures with more than 16 colors must be imported using the "*VGA*" loaders. "*VGA320*" accepts only 320x200 or 160x200 pixels.

Printfox/Pagefox

Format: Hires, 320x200/640x400/up to 640x800, compressed
Displayed Info: PFox, 320x200, Color
Loaders: ldr.PFoxResample
ldr.PFoxSelect
Savers: svr.PFox.GB
svr.Pagefox.90
Notes: Every Scanntronik graphics application is capable of producing two main output file formats: a 640x400 "*GB*" file, and a clip with variable sizes up to 640x800 pixels. With the "*Select*" loader you may freely place a 320x200 pixel "window" over a representation of the source file's dimensions. In addition, "*GB*" files are downsized to 320x200 pixels using the resampling method while imported by the appropriate loader.

Printtechnik Digitizer

Format: 4Bit
Displayed Info: GoDotDigi
PTDigi, 320x200, Color
Loaders: ldr.PTDigiI
ldr.PTDigiII
Savers: none
Notes: There are two loader modules for this video frame grabber. The hardware is accessed directly without further options. The use of a b/w video source is recommended to avoid interference effects.

Scanntronik Digitizer

Format: 4Bit
Displayed Info: GoDotDigi
ScTDigi, 320x200, Color
Loaders: ldr.ScTDigitizer
Savers: none
Notes: This hardware module has built-in options to adjust the input quality

and comes with a set of colored filters to produce color pictures. The GoDot software provides a different method, giving you the ability to assemble three separate 4Bit files (Red, Green and Blue) to produce a color picture with far superior quality. Exported to other platforms and using professional software can result in an image with up to 4096 colors, making your C64 a powerful image digitizing station.

SnapShot

Format:	4Bit
Displayed Info:	n/a
Loaders:	ldr..SnapShot
Savers:	n/a
Notes:	When you have just finished programming your killer application module for GoDot you definitely need one or more pictures of the user interface to illustrate your manual! With the snapshot loader installed (REU required) you are able to grab the current screen via (COMMODORE) + (G). Don't attempt to start this process during disk access. You can exit the snapshot function using the left arrow key.

Starpainter 128

Format:	Hires up to 640x720, uncompressed
Displayed Info:	StrPntr, 320x200, Color
Loaders:	ldr.StarPntr.128
Savers:	none
Notes:	Resampling technique is used to convert these 640x400 monochrome pictures to the internal data format.

VBM

Format:	Hires up to 640x400, uncompressed (v2) or compressed (v3)
Displayed Info:	ACE-VBM, 320x200, Color
Loaders:	ldr.VBM
Savers:	svr.VBM
Notes:	Like ldr.PFoxGB, this loader resamples the data to 320x200 pixels, either in 5 shades of gray (***Resample to Hires***) or in 9 shades of gray (***Resample to Multi***). The third option reads the picture as it is (***Monochrome***). You can't set clips in this loader yet, so what you'll get is the upper left corner of the image. You can get rid of horizontal stripes using *mod.MCScreenFix*. The saver makes uncompressed version 2 VBMs of 127 disk blocks. Don't forget to set the dither mode before saving.

SECTION 5

PROCESSING MODULES

Most of the modules are used to manipulate the 4Bit graphics data in GoDot's memory. You may enhance picture quality, perform various effects, combine pictures, change dimensions, colors and much more.

Since GoDot looks at image data as a gray scale image, special modules are necessary to perform color manipulations; in this section these modules are indicated with a (C). Other modules are used to enhance the GoDot user interface and support actions aimed to various peripherals. These "system" modules have an extra period between the mod. designator and the specific module file name.

In addition to system and manipulation modules, there are also printer driver modules. These are identified by a .drv extension at the end of the file name.

At this writing, there are many modules that are in a constant state of development. To keep up to date, you can check out what's new on the GoDot web site (*http://members.aol.com/howtogodot/welcome.htm*) or Email to *god.ADettke@t-online.de*.

System Modules

mod..ChangeDir

This module lets you change your working directory to subdirectories on 1581 floppy drives and CMD storage devices. Just choose one of the directory names from the *File Requester* and double-click it, or click once on the name and select the ***ChangeDir*** gadget. To return to the main directory of the device, click on the ***Root*** gadget.

mod..CMD-Clock

Owners of CMD devices that contain a built-in Real-Time Clock (RTC) can use this module to display the current time.

mod..ColorCount

Counts all C64 colors appearing in the current picture. The amount of color combinations as produced by the special IFLI/Plus4 prep function is shown separately, but is only relevant if the picture is in fact a multicolor type.

mod..FileCopy

Use this module to copy files of any length from one installed device to another (at least two drives are necessary—single drive copying is not supported). As file data is transferred block for block, the image data remains unaltered.

The user interface contains source and a destination areas, each with its very own device requester, a ***Scroll*** gadget and a ***?*** gadget to force error status information.

All actions are performed on the source device or a selected file on the source (left) side. When selected, a file name is marked inverse and displayed in the upper information area along with the file length. You may now ***Copy***, ***Move***, rename (***Name***), or delete (***Del***) the selected item by clicking the appropriate gadget. If you try to move a file, it won't be scratched from the source if writing to the destination fails.

The ***Dos*** gadget gives you the chance to manually perform disk commands. The information area is used for manual command input. The ***Quit*** gadget is used to exit from the module.

mod..FileType

Even though most image types have something in their name to help identify them, you may not be familiar with all types or which loader module to use. Select the name in this module's file requester, and it will try to identify the file type for you.

mod..FirstDrive

Some modules access the drive they are launched from when opening a file requester. Use this tool to change this behavior and force the selection of another drive.

mod..Histogram

A statistical tool that provides the number of different colored pixels in a specific image.

mod..HowMany

This module shows 1000 histograms, one for each 8x8 pixel block, and lets you see the theoretical information loss when rendering in the 4 color maximum provided by the VIC chip.

mod..NetCopy

This is a GoDot version of the "NetCopy" program for "64Net". The main window shows all files on the source device. Files appearing in the horizontal chooser gadget (in the middle of the window) may be selected by clicking on the name. Selected files are marked in reverse, and the total number of items is shown in a separate gadget. You may select or deselect all files via the ***All*** and ***None*** gadgets on the left, while the gadgets on the right let you jump to the start or end of the file list, scroll up or down, and also let you flip through the names page by page. The source is changed by a tiny gadget in the lower left corner (watch the little arrow!) either to a C64 drive or a path on the PC side (which must be typed in manually and completed with a slash). Note: the ***NCom*** gadget has no function yet; it is reserved to later implement installation of *.D64* image files as devices.

mod..NewPointer

Using this module, you may install your own pointers for GoDot. A file with a *PTR.* prefix contains the data of a typical sprite (63 bytes), plus two additional bytes that contain the position of the hot spot (the part of the pointer that triggers an action, which defaults to: x=0, y=0).

mod..PatternEd

This module is used to change the pattern used by the ***Dith:*** *Pattern* setting in the *Color Controls* area of the main screen. The large ***Pattern*** gadget at the top shows all current patterns assigned to the 16 gray scales and the corresponding colors. To edit a pattern, click on it, and it will appear magnified in the ***Edit*** gadget on the left. To have full visible control of the changes, a ***Preview*** gadget on the right shows the pattern in real size. The default drawing mode is *Dot* (as shown in the Mode gadget) and clicking in the ***Edit*** gadget toggles a dot from set to not set. ***Clr*** clears the contents of the ***Edit*** gadget, and ***Fill*** sets all dots. If you want all 16 patterns cleared, just select ***Clr*** and ***⇨All*** to copy the cleared ***Edit*** gadget into all patterns.

Any single change (prior to clicking ***Ok***) may be undone by clicking on ***Undo***, while ***Oh no!*** cancels all changes made to the current pattern.

The ***Ok*** gadget ends the editing process and transfers the pattern from the ***Edit*** gadget to its position on the upper ***Pattern Bar*** gadget. To use the same pattern as source for further modification in another position, use ***Copy*** and select a destination position. You may also copy the pattern from the Edit gadget into all of the patterns at once by clicking on the ***⇨All*** gadget.

A horizontal mirror image of the pattern can be created by selecting the ***Mirror*** gadget, while the ***Turn*** gadget will produce a vertical mirror image of the pattern. You may also use the ***Rotate*** gadget to rotate the pattern clockwise in 90 degree steps.

The ***Scroll*** gadget selects *Scroll Mode*, which lets you select the scroll the pattern in any of four directions by clicking along the inside edge of the ***Edit*** gadget. There are also ***HLin*** (*Horizontal Line Mode*) and ***Vlin*** (*Vertical Line Mode*) gadgets for drawing horizontal or vertical lines. Clicking on the ***Mode*** gadget (near the upper right corner of the ***Edit*** gadget) puts you back into *Dot Mode*.

You may save your new patterns using the ***Save*** gadget, while the ***Load*** gadget is used to load a previously saved pattern file. The ***Cancel*** gadget exits from the module without making any changes. The ***Accept*** gadget will transfer the new pattern data to the system for immediate usage.

If you have made pattern changes and want to get the default pattern back, clicking on the ***SysPat*** gadget will do this.

mod..REUTool

The main purpose of this module is to install additional modules into an REU. In addition, this module provides information on current REU usage and displays the next address available (bank and offset). A special gadget shows the REU size as determined at system start-up. *Note: GoDot recognizes a maximum of one megabyte.*

mod..Toasters

A screen blanking utility with a well known output. Now you don't have to switch off your monitor!

mod..VDCEd

C-128 users can use this tool to modify the registers of the VDC chip. The ***SLace*** gadget starts a special software interlace mode using 640x400 pixels on a 64K VDC. **Warning: We are not responsible for any damage to your monitor as a result of using this module, so use it at your own risk!**

mod..Version

This module displays the version numbers of the currently installed loader and saver. There is also a *ldr.Version* file that provides version numbers for loaded modules and the graphic render routines. If you prefer printed information use the *pvers* program to print the version numbers of all system modules located on a specified disk.

mod.MineSweeper

The object is to find all of the squares which do not contain a mine. A square may contain nothing, a mine, or a number that indicates how many mines are located in adjoining squares. In the upper left corner you'll see the number of squares you can click on without being blown up .

There are two reasons for the existence of this module. First: a good game at the right time can't be wrong. And second: GoDot is able to act as a universal front end for many applications. We invite all programmers to take advantage of this.

READ ME!

Whenever you see this on a GoDot disk it will provide you with information about the contents of the disk. This file is often used when updates have been made that are not covered in the manual.

Generic Modules

mod.3D-MosaicMC (C)

Every 8x8 pixel block of the image is filled with the predominant color. A 3-D effect is achieved by making the upper and left borders at a brighter level and the lower and right ones in a darker level.

mod.4Bit2PFox

The settings made in the *Screen Controls* and *Color Controls* sections of the main screen are used by this module as it renders an image into the free RAM of a Pagefox hardware cartridge. With properly installed launching software you can jump directly into Pagefox-compatible software. Simply copy *Eddifox*, *Handyfox* or *Digifox* to the GoDot disk and rename it *gofox*. Don't be alarmed by strange effects that take place while jumping; GoDot will survive, and COMMODORE + V will take you back safely.

mod.4Bit2VDC1 & mod.4Bit2VDC2

The settings made in the *Screen Controls* and *Color Controls* sections of the main screen are used by these modules as they renders an image in b/w on an RGB display monitor connected to the RGBI (80-column) output of a Commodore 128. Version 1 (*mod.4Bit2VDC1*) supports the 16K VDC (640x200 pixels), while version 2 uses a 640x400 pixel display with the 64K VDC (128D or upgraded 128).

mod.4BitLens

This provides a 40x24 pixel magnified display of the GoDot 4Bit data (not the rendered image). Use the cursor keys to move, HOME to jump to the upper left corner and * to find the center of the picture. If you rendered the current data, then it is useful to display the whole picture with the ((open parenthesis) key.

mod.ApplyColors (C)

The 4Bit values will be replaced by colors according to the current palette settings. This is an easy way to remove undesired colors from an image.

mod.ApplyDither (G)

This module forces a reduction of gray scales based on the dither settings, the resolution mode and the number of colors. The predefined color palette's settings are also used.

mod.ApplyMap (G) & mod.ApplyMapCol (C)

Transfers the settings of the balancing requester to the 4Bit data.

mod.AverageColor (C)

Use this module to perform Hires to Multicolor conversion. It converts neighboring pixels to a double-pixel, with the color coming from a mixing procedure.

mod.AverageGray (G)

This module is similar to *mod.AverageColor*, but determines the new double-pixel's gray value by taking an exact arithmetic average of the two source pixels.

mod.ClipWorks

This is the default module installed by GoDot if no other module has been specified. It is used to set the values belonging to the *Exec Area* and lets you set up a *clip section* on the screen via the ***Clip*** gadget (one click for the upper left and another for the lower right corner). The minimum size for a clip is one 8x8 pixel block, the maximum is 40x25 blocks. The current settings are displayed in the requester (Row, Column, Horizontal width, Vertical width) and may be modified manually. The ***Show*** gadget redisplays the image with a flashing clip area. Selecting ***Last*** restores the previous clip parameters. Toggling to ***Full*** will select the entire image as a clip. This procedure can be very useful when rendering in multi mode, because no new background color is calculated while in clip mode.

Additional functions are ***Zoom*** and ***Shrink*** for smooth resizing in gray scale mode. If you have to use real colors, use the "non-smoothing" *mod.Squeeze2Clip* and *mod.StretchClip* instead.

ClrClip fills the inside or outside of a clip with a specified color.

mod.DecodeGIF

This is the decoding section for use with the GIF loader module. Due to the quite sophisticated and memory hungry decoding routines the loader section provides the user interface and then invokes the decoding procedures located in this module.

mod.DynaRange (G)

A picture containing no or few dark and/or bright colors appears to have a flat dynamic. This module stretches the available dynamic range from black to white.

Error Distribution

mod.EDgray (G) & mod.EDgrayMC (G)

A gray scale reduction is performed using the error distribution method resulting in a non-regular pattern according to the current number of gray scales set in the palette requester. The dynamic appearance is maintained as well as possible. To minimize undesired effects it is recommended that you have at least black and white in the palette. The "MC" version uses multicolor mode double pixels.

mod.Edmono (G)

Performs a reduction to black and white by using error distribution without gamma correction. A b/w printout of an image produced without gamma correction often shows problems with middle gray scales. Setting a higher contrast level may solve this, but will also affect bright and dark gray scales. *Note: Gamma correction provides a nonlinear emphasis of the gray scales in the middle range.*

mod.Edmono-g0.35 (G)

Low gamma level for printer output.

mod.Edmono-g1.8 (G)

High gamma level for screen output.

mod.FlickerFixer (C)

Performs a special screening procedure to reduce flickering in interlace FLI mode.

mod.Gradient (G)

Uses your current image to create a new image with a "raised" look (be sure to save your image first if you want to keep it). Use the *Screen Controls* and *Color Controls* to set the parameters to generate smooth vertical transitions from one color to another. This can be used as backdrop for composing spectacular title pictures.

mod.MCScreenFix

This module swaps the nibbles of every second data byte to prevent a hires pattern from appearing as horizontal lines in multicolor mode.

mod.Mirror

Rotates the entire image around the vertical (y) axis.

mod.Negative (G)

Inverts all gray scales to produce a photo-negative effect.

mod.OddSwap

Just swaps the nibbles of all data bytes in every second line of the picture and removes a pixel pattern in pictures, preparing for better standard screen output.

mod.Quick4 & mod.Quick4TED

High speed rendering process using only 4 gray scales, without displaying the picture (use ***Redisp***). The TED version works with images imported via the *ldr.PCXprep4PI4* loader module.

mod.RasterIFLI

The two existing modules *mod.OddSwap* and *mod.MCScreenFix* are combined to enhance IFLI prepped images and imported GIF images for standard screen output.

mod.Rendered2Raw (C)

To export a rendered picture in a foreign file format it must be recalculated to standard 4Bit. All standard C64 saver modules use the rendered picture as output source!

mod.ScrollR8

As standard FLI is unable to display the right-most 3 character columns. This module provides an easy way to shift the important portion of the image to the viewable area.

mod.ShowVDCFloyd

Uses error distribution method to render the current 4Bit image buffer to the VDC screen in black and white.

mod.Squeeze2Clip (C)

As opposed to the Shrink function of the default ClipWorks module, this module resizes the image to clip size without changing the colors by simply skipping pixels and lines.

mod.StretchClip (C)

A color version of the Zoom function in the ClipWorks requester. The clip area is blown up to full screen size.

mod.Tile (G)

As a clip may only be aligned to VIC character blocks, it is impossible to reduce the image to an exact quarter. Use this utility to perform this and fill the screen with the clip. A Kaleidoscope effect is sometimes possible if used several times.

mod.TileClip

The whole picture is filled with the selected clip area. You may set the center of the clip or its lower right corner to the screen center and choose an x and/or y mirror function.

mod.Tint (C)

With this module we try to mix every color of the picture with a color of your choice. Due to the VIC's color limitations, not all combinations are useful.

mod.UpsideDown

Performs a 180 degree rotation around the z-axis.

mod.VSmooth (G) & mod.VSmoothCol (C)

Basically the same function as *mod.AverageGray* and *mod.AverageColor*, but works in the vertical direction.

Convolution Modules

Processing data through convolution is a common technique used to emphasize significant areas and enhance visual output. Basically, convolution works by comparing a specific data value with others. In GoDot, the data consists of a pixel and its surrounding neighbors. This area makes up a matrix in which various arithmetical, logical or statistical operations are performed. Each pixel of the matrix is assigned its own "weight", and the result of these operations may also be passed through filtering to avoid extreme effects. The values comprising the matrix may be varied, but the process of determining the matrix to produce specific results requires either much experimentation or a working knowledge of image processing theory.

The general purpose operators are flexible but limited to arithmetic functions. To provide other popular and well proven image processing effects we prepared a range of special operators optimized for speed and ease to use.

mod.Blur High (G)

Produces a heavy blurring effect.

mod.Blur Low (G)

Produces a more subtle blurring effect.

mod.Cartoon (C)

All colored areas are cleaned up and surrounded with a black border. Nice effect!

mod.CleanUp (C)

Removes a pixel if none of the neighbors has the same color.

mod.Convolve (G)

This module acts as general purpose convolution filter which can apply 3x3 convolution matrices to the 4Bit data. A number of presets are available, and your own creations may be saved to disk. The large matrix requester shows 3x3 cells, the center is called *kernel*. Activate a cell by clicking in it and enter a new value (-128 to 127) via the keyboard. The ***Shift*** gadget raises the image data values before passing them to the convolve operation. The ***Threshold*** gadget prevents all data with lower values from being processed. On the output side, the ***Mix*** gadget controls the level of combining the new values with the old ones; 100% leads to a full replacement of the old data. Gadgets are provided to ***Reset*** the matrix to zero and to ***Rotate*** it around the kernel. Please note that you may ***Save*** and ***Load*** Convolve module settings; three such files have been included on the distribution disk: *con.Emboss High*, *con.MakeSmoother* and *con.MakeDarker*.

mod.Convolve7Bit (G)

Use this tool to apply convolutions to image data in TED 7-Bit format (imported via the *ldr.PCXprep4Pl4* loader).

mod.CrossHatch (G)

Applies a fine texture to the picture.

mod.DeepPress (G)

All objects look like they are pressed into metal.

mod.Emboss (G)

A very popular relief effect , changes every picture to the surface of a coin.

mod.Gaussian (G)

Less blurring than the special blur operators.

mod.LaPlacian (G)

All areas loose their colors, only the borders are left.

mod.MedianFilter (C)

A useful tool to eliminate unwanted noise in digitized images.

mod.MedianFilter(TED) (C)

Performs the process on 7-Bit TED data imported by the PCXprep4Pl4 loader.

mod.MosaikMaxi (C)

Replaces a pixel with the brightest color of the neighborhood. The mosaic effect results due to the low resolution of the VIC.

mod.MotionBlur(Col) (G/C)

Uses the width value of the current clip to set the amount of smearing the colors to the left and to the right.

mod.OutlineColor (C)

Similar to mod.LaPlacian, but with colored borders. The less details, the better the result.

mod.OutlineMono (G)

Similar to mod.LaPlacian, but with white borders. Use it in conjunction with mod.Negative to produce mask images.

mod.RandMoveLeft (C)

Moves the pixels to the left up to a specified distance. If used several times the whole image is blown out of the screen.

mod.RandOblique (G)

Smears the colors in diagonal direction, thus simulating a brush effect.

mod.RemIsoPixels (C)

Replaces a pixel if all neighbors are of the same color.

mod.Sharpen1(3) (G)

Puts more accents on borders to enhance sharpness.

mod.Smear (C)

Randomly swaps pixel values.

mod.Speckle (G)

Similar to mod.Smear but with more significant results.

mod.WaterColor (C) & mod.WaterColor.O (C)

Simulates the use of water colors, preferring the most repeated color of the surroundings. The Oblique version (*mod.WaterColor.O*) adds a diagonal brush effect.

SECTION 6

PRINTER DRIVERS

All GoDot printer drivers support three types of interfaces: Wieseman, Merlin C+ and Centronics, the latter heavily recommended because of the highest printing speed. Printer drivers are loaded and executed just like other GoDot modules.

24Pin.drv

This module was written by Markus Kanet for a Stylus 800+ Epson printer, but works fine with all standard 24 pin printers. An option to set the linefeed is provided.

9Pin.drv

All Epson-compatible 9-pin printers are supported in 80 dpi and 240 dpi mode. As the output size is actually the same, 240 dpi mode is recommended for use with older color ribbons.

CanonBJC.drv

The only feature difference between this module and the *mod.HP.drv* is the ***Print Passes:*** *SLOW* option, which provides an 8 second break to allow the ink to dry when printing a transparency. This driver uses 360 dpi, so there is also a difference in the number of images that can fit on a page: 21 for *tiny*, 8 for *normal* and 3 for large (*lrge*). You may also center the printout by enabling the (SHIFTLOCK) key.

HP.drv

A color printer driver for all printers using PCL3. This covers all HP5xx printers, and also works with many laser printers and other compatible models.

The option gadgets are designed with the features of the HP560C in mind, but are switchable to generic HP5xx (b/w) and the basic HP500C (color) mode. Using a fixed 300 dpi mode, three output sizes are available from the ***Format*** gadget: *tiny* (up to 18 images per page), *norm* (3) and *lrge* (2). Since the HP5xx series are page printers, the sheet must be reinserted for additional pictures.

Print Quality may be modified choosing *low*, *high* or *KPad* (the latter lets you select quality via the printer's keypad). The driver can adjust ***Ink Usage %*** (settings preceded by a *y* indicate the use of internal gamma correction), and can also control the number of ***Print Passes***.

When using a color printer without black as a separate color (DJ500C) it is recommended that you use the ***Balanced Black*** mode, which helps to correct the somewhat olive appearance of the black emulation.

Finally, the ***Position*** gadget lets you choose between 18 different output positions.

StylusC.drv

The Epson Stylus Color drivers works like the Canon driver, but has no ability to control ink usage.

SECTION 7

TUTORIALS

Tutorial 1: Composing An Image

Before starting, REU owners may wish to install the modules we'll be using into their REU (using the module *.REUTool*).

To prepare the background image ***Load:*** the *IFF* loader and then ***Load*** the image file *Clouds3.IFF* using ***Gray Scale*** mode.

REU users are advised to save a backup copy to RAM to avoid a second disk load later on. Just open the *File Requester* (click on ***Save***), choose ***Unit RAM***, select ***Undo 4Bit***, then click on ***Save***.

Our first job is to apply a more natural looking palette to the gray scale image. Unfortunately, there are only two shades of blue along with cyan. It's an easy task for GoDot, though. Click on ***Inst*** in the *Image Operators* area, select the *ApplyDither* image operator, then click on ***Load***. When you return to the main screen, click on ***Execute***. Now set the following options in the requester:

Dith: *Pattern*
Mode: *Multi*
Reduce to 6 Cols

Click on ***Accept***. To choose the colors for the image set ***Colors*** in the *Screen Controls* to ***6***, then click on ***Palette*** in the *Color Controls*. Select ***Default*** in the *Palette* requester, then replace the dark gray with the dark blue (click on the dark blue block in the *Source Palette* at the top, then click on the dark gray block in the *Destination Palette* at the bottom). Now click on ***Accept*** to activate the changes and exit back to the main screen. Next we need to transfer the colors of the current palette to the 6 gray scales that our image was reduced to. The image operator *ApplyColors* will do that for us. Install (***Inst***) the *ApplyColors* module and ***Execute*** it. Now set the ***Colors*** in ***Screen Controls*** back to *16*, and reset to the default palette by clicking on ***Palette*** in the *Color Controls*, then select ***Default*** and ***Accept***. You many now render the image to screen by clicking on the ***Display*** gadget. You now see the initial results: a clouded sky in shades of blue and green.

Before we move on, ***Save*** this part of the work to disk with the name *Back.4Bit* using the *4BitGoDot* saver. **Note:** Users with an REU may save the file using the ***Temp*** saver. To do this, load the ***Temp*** saver by clicking on the gadget next to ***Save:***, select ***Temp*** from the *File Requester* and click on ***Load***. Now click on ***Save*** and select ***RAM*** from the ***Units*** window. Enter ***temp00*** in the *Input* window of the *File Requester* and click on ***Execute*** to bring up the *Save Temp* requester. Make sure the top gadget of this requester shows the correct RAM device type (you can adjust it by clicking on it if it doesn't), then click on ***Save***. You may exit the module by clicking on ***Leave***.

The next step is to reload the *Clouds3.IFF* image (REU users can select *Undo4Bit* from the *Unit RAM*).

Next we'll use the loader *4Bit&Map* to place a foreground over the existing image. Click on the ***Load:*** gadget and select the *4Bit&Map* loader from the *File List*, then click on *Load*. When you return to the main screen, click on the *Load* gadget. Clicking on the ***Modify Data*** gadget in the requester brings us to the composing section of the module. Now click on the Alpha Channel gadget (***AlphaCh***), first making sure that the gadget next to it has the ***pos*** option showing. Select the *SphGodAlpha.4Bit* file from disk, and click on *Load*. This file is an image of the word "GoDot" wrapped around a circular object. This image will be placed over the background, and as a side-effect will lower the overall brightness level of the rest of the image.

Next we'll use a bump-mapping technique to add realistic lighting effects and make the circle appear to be a sphere. To do this we'll continue to use the 4Bit&Map loader, so start by clicking on *Load*. Click on the ***Modify Data*** gadget, then select the ***Bump Map*** function. In the *File Requester* choose the *SphereBump1.4Bit* file and click on *Load*. Shades and light reflections have now been added to produce a translucent sphere with text wrapped around it.

Now comes the nifty part: we're about to put the blue background right behind the red sphere! Click the gadget next to ***Load:***, select the ***4Bit&Mask*** loader from the *File List* and click on *Load*. Click on the ***Get Mask*** option (make sure the ***from*** gadget next to it is set for *Disk*). Choose *Sphere.Mask* (this is a monochrome bitmap image) from the *File List* and click on *Load*. We can take a look at this by clicking on the *View Mask* gadget to see a white circle on a black background.

Now we want to reload the background image we saved earlier, using the white areas of the mask to protect portions of the current image. If you have an REU and saved the background image as *temp00* earlier, set the ***from*** gadget next to the ***Get 4Bit*** gadget to *Temp*; otherwise leave it as *Disk*. Now click on the ***Get 4Bit*** gadget. If you're getting the file from disk, select the *Back.4Bit* file from the *File List*, and click on *Load*.

Now click on ***Leave*** to exit from the loader and render the entire new image with the ***Display*** gadget. After the full image has been rendered, click once (anywhere) to exit back to the main menu.

To complete the project, we'll add one final touch by overlaying some text onto our image. Click the gadget next to ***Load:***, select the *4BitGoDot* loader, then click on the *File Requester Load* gadget. Now click on the Load gadget in the ***Commands*** area. In the Loader requester click on the ***Compose*** gadget to bring up the *Compose* requester. Now click on the ***Foreground*** gadget. Select *IP-2.4Bit* from the *File List* and click on *Load*.

With the appropriate saver module we may export the work to our favorite paint application.

If our main purpose was to prepare the image for printer output we should have chosen *Hires* instead of *Multi* before reducing the colors because of the finer pattern, for output by a black and white printer even color reduction isn't necessary.

Tutorial 2: Using Clips

This tutorial assumes you are now familiar with the controls, and shows how to take advantage of having two image buffers: the 4Bit data and the rendered bitmap.

As in the first tutorial, we'll start by preparing a backdrop. To begin this process, *Load* the image file *Leather1.IFF* using the ***Gray Scale*** option of *ldr.IFF*.

Next we reduce the 16 gray scales to a 9 color palette with shades of brown (or at least with the VIC's impression of them). ***Load*** and ***Execute*** the *ApplyDither* image operator with following settings:

Dith: Pattern
Mode: Multi
Reduce to 9 cols

At this point you should ***Save*** this part of your work to disk using the name *Back1.4Bit* using *ldr.4BitGoDot.*

REU users are advised to save a backup copy to RAM to avoid a second load from disk. Just choose ***Unit RAM*** in the file requester and double click ***Undo 4Bit***.

Next, Load the file *Nagel01.IFF*, again using gray scales.

Now we're ready to scale the picture down to 50% and place it somewhere in the upper left corner of the screen. ***Load*** and ***Execute*** the *ClipWorks* module to do this. Perform a rough on-screen clip setting, and then fine-tune the values to:

Row: *3*
Col: *4*
Hor: *20*
Ver: *13*

Now use ***Show*** to view the clip area and verify the aspect.

The ***Shrink*** gadget does the rest. *Note: Since we're work with gray scales, this method (while somewhat slow) is recommended, while using mod.Squeeze2Clip is a quicker solution for color images.*

Leaving *ClipWorks* shows us a changed gadget on main screen: the ***Exec Area*** gadget is set to *Clip*. That is important, because rendering now only takes place in the selected area, leaving the previously rendered picture unharmed. At this time, selecting the ***Display*** gadget would open a window to the 4Bit data.

Now use the *FrameClip* module to apply a yellow border to the window and do again display the result.

To produce the effect of the Nagel motive floating over the backdrop we'll add a shadow by lowering the brightness of the affected area. This is accomplished by setting the ***Brightness*** level in the *Balancing* requester to *-4*.

Now Save this part of the work to disk using the name *Back2.4Bit* with *svr.4BitGoDot.*

REU users are advised to save a backup copy to ram to avoid a second load from disk. Just choose ***Unit RAM*** in the file requester and double click ***Undo Rendered*** this time.

The previously saved file, *Back1.4Bit* (or *Undo 4Bit* for REU users) has now to come back into the 4Bit buffer. ***ClipWorks*** is used to set the values for the shadows and perform an immediate rendering while the ***Exec Area*** remains set to *Clip*!

Settings of the lower part of the shadow:

Ver: *2*
Row: *16*
Col: *6*
Hor: *20*

Mind the order! And now the right side:

Row: *5*
Hor: *2*
Col: *24*
Ver: *11*

Rendered2Raw is the right tool to transfer the rendered image back into 4Bit.

The last thing to do is to paste some text information into the free space at the lower right. ***ColManOver.4Bit*** is a prepared colored overlay which we place with the *4BitGoDot* loader using the **Compose/Foreground** option.

And finally, be sure to reset the Balancing values and Clip Size before rendering.

SECTION 8

GODOT TIPS & TRICKS

Flipping an Image

To rotate an image around its horizontal (x) axis use both of the modules *mod.UpsideDown* and *mod.Mirror*, one after the other.

Creating Masks for ldr.4Bit&Mask

Images that have a distinct (e.g., single color) background can be easily transformed to masks. Before you begin, be sure to ***Save*** your image. Next, install (***Inst***) the PatternEd module (*mod..PatternEd*) and select the following options in order:

```
[ FILL ]
[⇨ ALL ]
[INVERT]
[ COPY ]
```

Now click the color that you do NOT want to be part of the mask (i.e. the one background color). Now click on:

```
[ACCEPT]
```

When you are returned to the main screen, set the following option in the *Color Controls* area, set the following option:

Dith: Pattern

In the *Screen Controls* area, set these options:

Resolution: *Hires*
Colors: 2

Now select ***Display***. GoDot will render a blue & black mask, which is what you'll need for *ldr.4Bit&Mask*. Be sure to ***Save*** the mask image!

Reconfiguring the REU

If you want to completely rebuild the contents of your REU, but when rebooting, GoDot seems to ignore all RAM commands in your *INI* file. The solution is to switch to the *File Requester* (click *Load*), select RAM from the *Unit* window, and reset your computer using the reset button (if you have one; if not try (STOP + RESTORE)). The next time you boot, all modules will be newly installed. Explanation: The command REUCLEAR=0 in the *GOD.INI* file prevents GoDot from deleting the REU directory on start-up (its flag resides *inside* the REU). When you activate REU access (unit RAM), GoDot exchanges RAM and the execution area, so the flag isn't in the REU anymore. GoDot can now launch "normally". If you set REUCLEAR=1 at the top of an INI, you would still need to boot twice for the change to take effect.

svr.PFoxGB Produces Blocky Images

All savers that create a 640x400 monochrome image (*svr.PFoxGB*, *svr.PCXmono*, *svr.GeoPaint*, *svr.Pagesetter*, *svr.Pagefox.90*, *svr.PCXmono.90*, *mod.4Bit2VDC*, *mod.4Bit2PFox*, *mod.9Pin.drv*, *mod.24Pin.drv*, and the black and white mode of the color printer drivers) work together with the settings of the *Color Controls* Area! Above all the dither settings will do miracles.

Files Can't be Loaded

There are three possible reasons for this:

a) The filename ends with one or more spaces. Remedy: Click the filename once, it appears in the input gadget. Click again there, type in an asterisk (*) next to the last character. Then load.

b) The name of the file contains non-PETASCII characters (e.g. GEOS names). Characters that appear to be uppercase actually are GEOS lower case letters. These characters indeed are part of the C64 character set and are displayed, but you can't produce them by typing directly on the keyboard. Remedy: Substitute a question mark for each uppercase letter . Then load. (In GEOS: use only uppercase letters to avoid these problems!)

c) The file does not appear in the directory although you just have saved it. Reason: You typed a period as a fourth letter. Remedy: Use *mod..FileCopy* to rename the file.

FileCopy Tips

a) Delete files: single files are deleted by clicking their name and then ***DEL***. Multiple files can be deleted manually by DOS. To delete all files beginning with the letter "a", Click DOS, type s:a* and press (RETURN). This procedure will also work in the file requester: click into the input gadget, type a* and (RETURN), then click ***Delete***. Caution: A single "*" will delete ALL files of the present prefix, e.g., all loaders!

b) Copy files: If you know the name of a given file and it is situated far at the end of a long directory, click any file. Its name will appear in the filename display at the top of the screen. Click there, delete this input by (COMMODORE) + (X), and then type in the name wanted. Click ***Copy*** after pressing (RETURN).

c) Copy files that already exist on the target drive: Lock the (SHIFTLOCK) key before hitting Copy. (Pressing (SHIFT) will do as well.)

PCX Pictures

If you import clipart (black and white pictures) from the PC, you should take care of their mode when saving on PC. They shouldn't be 256 color pictures at all, because *ldr.PCX-VGA320* expects images of exactly 320x200 pixels (respectively 160x200). If you saved them as two or 16 colored pictures you can read them with *ldr.PCX-EGA*, no matter what size they are. If the image only displays dark blue instead of white, activate the *palette* requester, set blue to white, and then start *mod.ApplyColors*. (Reset the palette after that.)

Files Can't be Saved

If GoDot stops the ***Save*** function without even starting to save, activate *mod..FileCopy*, set the target drive to the source drive, click whatever file you like, and then ***Copy***. Since copying is impossible on one drive, GoDot will stop with an "ERROR". Exiting *FileCopy* now will result in the ability to ***Save*** successfully. Explanation: Reading a file (for ***Copy***) moves the read/write head of the disk drive off the directory track. Recognition of the drive will work again after this.

DOS Function Refuses to NEW a Disk

Unformatted disks and disks from other platforms can irritate the floppy drive's recognition of disks, if you accessed them. Put another C64 formatted disk into the drive, let *FileCopy* display its directory, exchange the disks, and then try DOS (n:whatever name,id) again. It will work.

Umlauts via Keyboard

ä	(small a-umlaut)	SHIFT + ;
Ä	(big a-umlaut)	COMMODORE + A
ö	(small o-umlaut)	SHIFT + :
Ö	(big o-umlaut)	COMMODORE + O
ü	(small u-umlaut)	@
Ü	(big u-umlaut)	SHIFT + @
ß	(sz ligature)	£

Rendering Prepped Images

Color images that are loaded in standard multicolor, hires and even FLI formats needn't be "prepared" prior to printing to a color printer. However, images loaded with *ldr.GIF* and *ldr.PCXprep4IFLI* do need to be pre-processed because these loaders interpret the data so that the image can later be saved in IFLI format (*svr.FunPaintII*). IFLI is a format that appears to allow more than the 16 colors that actually exist on the C64. In reality, however, IFLI is simply two FLI images that are alternated in rapid succession, with one image being offset from the other by one pixel. To the human eye, which is unable to fully perceive that separate images are being alternated, the adjacent colors in the two images appear to mix and form new colors.

GoDot maintains the data from the two FLI images by combining the data in an alternating way into a single image. Within this 'master' image, the first pixel of each row of data is the first pixel from the first FLI image, the second pixel is the first pixel from the second image, the third pixel is the second pixel of the first image, and so on. This pattern of alternating the data from the two images continues throughout the entire master image, generating alternating columns (stripes) of one pixel width. The advantage of storing the data this way is that GoDot can render the image in hires or multicolor modes with less effort; the downside is that these stripes don't look good in a color printout. To mask the effect, prep the image using the module *mod.OddSwap*. This module will swap even and odd pixels in every other line. A color printout of the result will look quite natural, and having different colors close together will seemingly produce more colors than normal for a C64 image.

INI Files

GoDot obtains many of the settings for its defaults by reading the file *god.ini*. This file can be edited, or you may create other ini files with settings for screen colors, the mouse pointer, which drive to access first, and other program parameters.

To create or edit ini files, you use the BASIC editor. However, unlike BASIC programs, the line number is followed by a quotation mark ("), and each line may contain only one command. The available commands are listed in the INI FILE COMMAND SUMMARY, located after the following page. Be sure to save your file with a name that begins with the four letters *ini.* so that GoDot can recognize it. Also note that GoDot will ignore incorrect lines when executing the file.

To inform GoDot of this new INI file, you may add its name to the LOAD or RUN command as a parameter. For example, if you had a new ini file called ini.scan, you would do one of the following to have it loaded when you start GoDot:

```
LOAD "GODOT",8: "SCAN"  SHIFT + RUN STOP
```

or

```
LOAD "GODOT",8   RETURN
RUN: "SCAN"      RETURN
```

The following is an example of how the *god.ini* file looks when loaded into the BASIC editor and listed (in Uppercase/Lowercase mode to avoid graphic symbols!). It may help you in understanding how to create your own INI files.

```
120 "mousexy   =  300,160
150 "colors    =  16
160 "mode      =  1
210 "ram mod..FileCopy
220 "ram mod..REUTool
225 "ram ldr.4BitGoDot
230 "ram mod.ClipWorks
330 "ram mod.MineSweeper
340 "reuclear =  0
```

INI FILE COMMAND SUMMARY

Command	Parameter	Default Values	Value Limits	Purpose
clip	row, column, width, height	0, 0, 40, 25	0-24,0-39,1-40,1-25	preset a clip
colors	number	2	2-16	preset the number of rendering colors/shades
defcolors	# of colors, color1, ...color(#)	e.g.: 16, 0, 6, 9, 11, 2, 4, 8, 12, 14, 10, 5,15, 3, 7, 13, 1 (def.)	2-16, each 0-15	preset of one of the 14 default palettes
loader	ldr.xxxxxxx	ldr.4BitGodot	5-16 characters	installing a loader module
loadfrom	drive number	(boot drive)	8-11	first accessed drive when loading pictures
mode	number	0	0 or greater 0	0 = Hires; greater 0 = Multicolor
module	mod.xxxxxx	mod.ClipWorks	5-16 characters	installing an ip module
mousecolors	foreground, background	1, 0	0-15, 0-15	preset colors of mouse pointer
mousexy	x-position,y-position	80, 80	0-511, 0-255	preset starting position of mouse pointer
pattern	pat.xxxxxx	pat.System	5-16 characters	preset dither "Pattern"
pfox	number	1	0 or greater 0	0=Pagefox reserved for modules, non-zero=Pagefox for graphics
ram	xxx.xxxxxx	-	5-16 characters	installing module of any kind into a REU
reuclear	number	1	0 or greater 0	0= don't initialize REU on reboot, non-zero= initialize REU on reboot
saver	svr.xxxxxx	svr.4BitGodot	5-16 characters	installing a saver
saveto	drive number	(boot drive)	8-11	first accessed drive when saving pictures
textcolors	shadow, lite, normal text, hilite, background	0, 14, 12, 15, 6	0-15 for each parameter	preset main screen colors
ticks	number	13	0-255	preset counter for double click
vdc	number	1	0 or greater 0	0=VDC reserved for modules, non-zero=VDC for graphics

— USER NOTES —

GoDot

DISK 1

Side One: Main Program and Modules
Side Two: Additional Programs and Modules
See Manual For Loading Instructions

©1991-1997 by A. Dettke & W. Kling
Distributed in the United States by Creative Micro Designs, Inc.

GoDot

DISK 2

Side One: Sample Images
Side Two: Additional Sample Images
For Main Program, See Disk 1

©1991-1997 by A. Dettke & W. Kling
Distributed in the United States by Creative Micro Designs, Inc.